

Topology Preserving Compression of 2D Vector Fields

Suresh K. Lodha, Jose C. Renteria, and Krishna M. Roskin

Department of Computer Science

University of California, Santa Cruz, CA 95064

{lodha,renteria,krish}@cse.ucsc.edu

Abstract

We present an algorithm for compressing 2D vector fields that preserves topology. Our approach is to simplify the given data set using constrained clustering. We employ different types of global and local error metrics including the earth mover's distance metric to measure the degradation in topology as well as weighted magnitude and angular errors. As a result, we obtain precise error bounds in the compressed vector fields. Experiments with both analytic and simulated data sets are presented. Results indicate that one can obtain significant compression with low errors without losing topology information.

Keywords: compression, topology, vector fields, error metrics, clustering.

1 Introduction

Reducing the size of large data sets while preserving essential features is an important problem. A great deal of effort has gone into polygonal mesh reduction and compression algorithms for computer graphics applications, and to a limited extent, for scalar data sets. The problem of simplifying or compressing vector data has been addressed only recently [13, 6]. Typically, the end result of these algorithms is a simplified view of vector fields using streamlines, streamtubes, textured images or a superimposition of these techniques.

In this work, we focus on compression algorithms that tend to preserve vector field topology. Topology-based methods were introduced in vector field visualization by Helman and Hesselink [7]. These methods (i) extract critical points of vector fields, namely the zeros, (ii) classify critical points, and (iii) compute the streamlines emanating from saddle points, referred to as separatrices, and terminating at other critical points or edges. The separatrices allow a user to obtain a decomposition of the domain into mutually disjoint regions that are topologically equivalent to uniform flow. The topological visualization of the vector field presents a greatly simplified view of the vector field and often allow the user to grasp the behavior of the data set quickly.

This paper presents a bottom-up compression algorithm based on constrained clustering. The algorithm starts with the original data set and merges adjacent clusters that are closest to each other as measured by some error criteria. We use a combination of local and global error criteria. The local error metric is based on a weighted combination of magnitude and angular error. For a global error metric, intended to measure the degradation in topology, we utilize the earth mover's distance (EMD) metric introduced by Rubner et al. [10] and utilized for vector field comparisons by Lavin, Batra and Hesselink [9, 3]. Our approach allows us to obtain significant compression with low errors while preserving topology.

The structure of the paper is as follows. Section 2 describes related work. Section 3 discusses the constrained clustering algorithm, the error metrics and the impact of using different weights on compression ratio and topology preservation. We also briefly

describe our implementation. Section 5 presents results of our experiments with some analytic and simulated data sets. Finally, Section 6 summarizes the results and indicates some future directions.

2 Related Work

Computation and visualization of vector field topology was introduced by Helman and Hesselink [7]. In particular, they also discussed computation and classification of different types of critical points. Critical points are points in the velocity field where the velocity vector is zero. An index of a critical point is the number of rotations of the flow around a critical point [12]. A 1st-order critical point is a critical point with index +1 or -1. Classification of 1st-order critical points is based on the eigenvalues of the velocity gradient tensor or the Jacobian. Depending on the sign and the type (real or complex) of the eigenvalues, eight different types of critical points are identified – attracting node, repelling node, saddle point, attracting focus, repelling focus, attracting star, repelling star and center. Visualization of higher order critical points is discussed by Scheuermann et al. [11].

Although the advantages of topology visualization have been well recognized, there are some limitations as well. Scheuermann et al. point out that the standard topology visualization techniques may not do an adequate job of identifying and visualizing non-linear topology or higher-order critical points [12]. Kenwright has observed that some standard topology visualization techniques may miss other important features such as attachment and separation lines [8]. The topology preserving algorithm presented in this paper has been interfaced with the topology extraction code developed by de Leeuw in [4]. It is certainly possible to interface our compression algorithm with other topology generation algorithms that capture additional or higher-order features.

de Leeuw has argued in favor of simplifying (or not preserving) vector field topology because the standard topology visualization for a turbulent flow may have too many critical points resulting in a cluttered image that may be difficult to interpret [4]. Although in this work, we have focused on a topology preserving algorithm, our algorithm can be enhanced to incorporate error metrics that can remove critical points of lesser importance. We will discuss this later in Section 5.

Independent of the above-mentioned efforts for visualizing vector field topology, researchers have begun to investigate algorithms for compressing and simplifying large vector data sets. Telea and Wijk introduce a clustering algorithm that iteratively simplifies a vector data set by merging two clusters that are similar [13]. They also introduce a notion of similarity based on elliptic contours. The simplified data sets are visualized using arrows and spot noise textured background. Heckel et al. present an algorithm for hierarchical representation of vector fields using clustering and principal components analysis [6]. The compressed data sets are visualized using streamlines and streamsurfaces and compared against the original data sets. In this work, we are interested in exploring the

effect of compression on topology visualization.

To quantify the error introduced due to compression, many different types of error functions are commonly employed in scientific visualization [15]. In the last couple of years, Lavin, Batra and Hesselink have utilized a global error metric based on the topology of vector fields [9, 3]. The global error metric, referred to as the Earth Mover's Distance (EMD), was first introduced by Rubner and Tomasi [10] for fast retrieval of similar images in a large database. We describe the EMD metric later in Section 3. In this work, we use a weighted combination of local magnitude error, angular error, and topology-guided node importance error that we introduce later in Section 3.2. We also study the effect of the weights on the amount of compression and the preservation or destruction of topology.

Globus et al. [5] discuss 3D vector field topology. Bajaj et al. discuss visualization of topology of scalar data sets and topology preserving simplification of scalar data sets [1, 2].

3 Compression

We now begin by presenting an overview of our approach for a topology preserving compression algorithm. We refer to a compression algorithm as topology preserving if the original and the compressed data set have the same number and type of critical points at the same locations. It is certainly possible to relax or strengthen this definition in different ways. One way to relax the definition would be to allow small shifting in the locations of critical points. In this paper, we focus on only those algorithms which preserve the locations of the original critical points. Therefore, our algorithm can be viewed as giving a very high weight (infinite) to the error arising due to changes in the location of the critical points. One way to strengthen the definition would be to impose additional conditions (such as the end conditions) on the integral curves originating from the saddle points. In practice, we have found, as discussed later in the examples, that integral curves remain well-behaved (closely following integral curves in the original data set) for significant amount of compression. Yet further, one may require other features such as attachment and separation lines be preserved. Preservation of additional features is an extremely important area of research, but beyond the scope of this paper.

As a first pre-processing step, we compute the topology of the given data set. For this purpose, we simply use the topology generation code of de Leeuw [4]. In the second step, the user inputs the weights to construct a weighted error metric that is then used in the compression algorithm. Our program does provide default weights and we also discuss heuristics for choosing weights. Once the error metric is formulated, a bottom-up clustering algorithm is performed that iteratively simplifies the vector data set step-by-step until the last step when all the data points have exactly the same value – the average of all the values. In the third step, the user is presented with an interactive visual display of topology resulting from the compressed data set and several error graphs including global EMD error and local magnitude and angular errors at each step of the algorithm. The user can interactively choose and view the resulting topology at different compression level or error thresholds. The user can also enter alternative weights to initiate the compression algorithm with a new weighted error metric.

We now discuss each of these steps in detail. Section 3.1 describes the compression algorithm. Section 3.2 describes the error criteria currently offered in our algorithm. Section 3.3 describe our experience of using different weights.

3.1 Algorithm

The clustering algorithm used in this work is similar to the one used by Telea and van Wijk [13]. We use node-based clusters (rather than cell-based clusters) and a different error criterion. Our input is a 2D

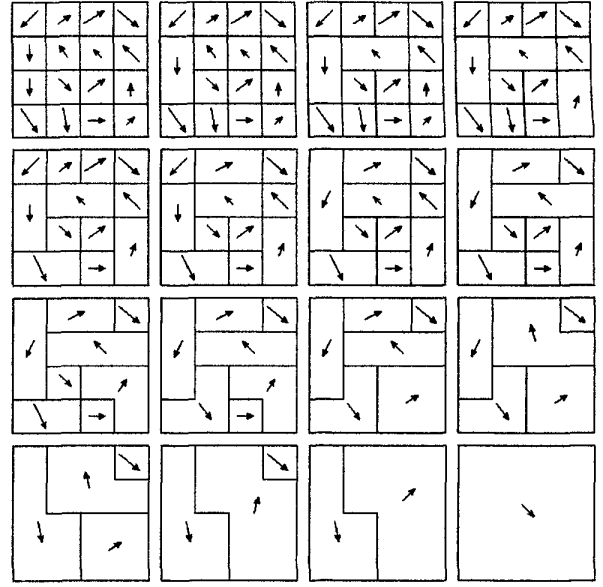


Figure 1: Compression algorithm applied to a 4x4 vector data set

vector field sampled on a rectilinear grid. Each node of the grid has a 2D vector associated with it. A cluster of nodes is a contiguous set of nodes with same vector data associated with all the nodes in the set.

As the first initialization step, each node belongs to its own cluster. For every node, the cost (as defined by the error criteria) of merging it to any of its neighbors is computed. The neighbor with the minimum associated cost is the preferred one for merging. The node which has the minimum associated cost of merging to its preferred neighbor is merged to its neighbor to form a cluster consisting of two points. The vector data associated with the new cluster is the average of all the vectors in the cluster. For all the nodes neighboring the cluster, the cost of merging the nodes to the new cluster is recomputed.

The above process is now repeated one step at a time to find the two clusters with minimum associated cost of merging. These two clusters are merged. The vector associated to the new cluster is the average of the two clusters. That is, if N_1 and N_2 are the number of nodes in the two clusters represented by the vectors \vec{v}_1 and \vec{v}_2 , then the new cluster has $N_1 + N_2$ points and the associated vector \vec{v} is

$$\vec{v} = \frac{N_1 \vec{v}_1 + N_2 \vec{v}_2}{N_1 + N_2}.$$

Finally, the cost of merging the neighbors of these two clusters is updated. This compression step is repeated $N - 1$ times, where N is the total number of data points.

Figure 1 illustrates the process of compressing a 4x4 grid of vector data set in 15 steps using the weights (described later in Section 3.3) $w_1 = 1$, $w_2 = 1$ and $w_3 = 0$, that is all nodes are treated identically ($w_3 = 0$) and the local magnitude error and the local angular errors are given equal weights ($w_1 = w_2$).

3.2 Error Criteria

In this work, we use three types of error – global EMD error, local magnitude and angular error, and node importance error. The final error is a weighted combination of these three types of error. The

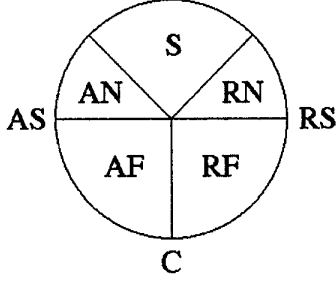


Figure 2: Classification of critical points according to their phase plane coordinates: S (saddle point), AN (attracting node), RN (repelling node), AF (attracting focus), RF (repelling focus), AS (attracting star), RS (repelling star), C (center).

weights can be provided by the user. We now discuss each of these three types of error.

EMD Error: The global EMD error is a way to compute the similarity or dissimilarity between the topology of two vector fields – the lower the EMD, the more similar the topology of two vector fields. The EMD between two vector fields depends upon three parameters: (i) number of critical points, (ii) type of critical points, and (iii) the eigenvalues associated with each critical point.

Let V_1 and V_2 be two vector fields. Let n_1 and n_2 be the number of critical points in V_1 and V_2 respectively. Each critical point will be mapped to a circle using a *phase plane* mapping as follows. Let λ_1 and λ_2 be the two eigenvalues of a critical point. Let $P = \lambda_1 + \lambda_2$ and $Q = \lambda_1 \times \lambda_2$. Let $\hat{\alpha} = P$ and $\hat{\beta} = \text{sign}(P^2 - 4Q)\sqrt{|P^2 - 4Q|}$. Finally, let,

$$\alpha = \frac{\hat{\alpha}}{\sqrt{\hat{\alpha}^2 + \hat{\beta}^2}},$$

$$\beta = \frac{\hat{\beta}}{\sqrt{\hat{\alpha}^2 + \hat{\beta}^2}}.$$

The mapping (λ_1, λ_2) to the unit circle (α, β) is referred to as the phase plane mapping. We will refer to (α, β) as the phase plane coordinates of the critical points. This mapping is shown in Figure 2. Critical points can be classified depending upon their phase plane coordinates [9], as shown in Figure 2.

Let $\{(\alpha_{11}, \beta_{11}), \dots, (\alpha_{1,n_1}, \beta_{1,n_1})\}$ and $\{(\alpha_{21}, \beta_{21}), \dots, (\alpha_{2,n_2}, \beta_{2,n_2})\}$ be the phase plane coordinates of the critical points of the vector fields V_1 and V_2 respectively. Without loss of generality, assume that $n_1 < n_2$. We add $n_2 - n_1$ points at the center of the unit circle with the phase plane coordinates $(0, 0)$ for the vector field V_1 . In other words, $(\alpha_{1i}, \beta_{1i}) = (0, 0)$ for $(i = n_1 + 1, \dots, n_2)$. The rationale for doing this is that a point at the center corresponds to a regular point, that is, not a critical point. With this modification, the two vector fields have now the same number of points, namely n_2 points in the associated lists. Now, let us set up a one-to-one correspondence between the two sets of points. Let γ denote such a correspondence between the two sets. Let Γ be the set of all possible correspondences between the two sets. Let

$$c_\gamma = \sum_{\gamma} \sqrt{(\alpha_{1i} - \alpha_{2j})^2 + (\beta_{1i} - \beta_{2j})^2}.$$

That is, c_γ is the sum of the Euclidean distances between the corresponding points. The EMD between the two vector fields is now defined to be the minimum c_γ amongst all possible correspondences between the two sets. That is,

$$EMD(V_1, V_2) = \min_{\gamma \in \Gamma} c_\gamma.$$

Conceptually, EMD is the minimum amount of work that must be performed to move one set of points to another.

Local Magnitude and Angular Error: Local magnitude error M and angular error θ between two vectors \vec{v}_1 and \vec{v}_2 are simply defined as $M = \text{magnitude}(\vec{v}_1 - \vec{v}_2)$, and $\theta = \text{angle}(\vec{v}_1, \vec{v}_2)$.

Node Importance Error: This error I can be associated with a node depending upon the importance of preserving a node. Since we allow user-defined weights for each kind of error, one can assume that this error can be specified to lie between 0 and 1. In this work, for topology preserving compression algorithms, we assign the importance to be 1 for critical points and its neighbors and assign the importance as 0 for all the remaining points. If I is chosen to be zero, then the compression algorithm reduces to an ordinary compression algorithm that does not take into account the topology information. Later we will compare the performance of our topology preserving compression algorithms with the non-preserving algorithms.

3.3 Effect of Weights

Weighted Error: The weighted error E for the compression algorithm is

$$E = w_1 M + w_2 \theta + w_3 I,$$

where w_1 , w_2 and w_3 are the weights associated with the local magnitude error, local angular error and the node importance error respectively. For topology preserving compression algorithms, we choose very high weight w_3 (typically 10,000) to ensure that the critical points are preserved a long way in the compression algorithm.

We illustrate the use of weights by presenting an example from the following data set. The data set we use is the gradient flow field associated with the following height field on the unit square:

$$f(x, y) = \frac{3}{4}e^{-\frac{(9x-2)^2 + (9y-2)^2}{4}} + \frac{1}{2}e^{-\frac{(9x-7)^2 + (9y-3)^2}{4}} + \frac{1}{5}e^{-\frac{(9x-4)^2 + (9y-7)^2}{4}} + \frac{3}{4}e^{-\frac{(9x+1)^2}{49} - \frac{(9y+1)^2}{10}}.$$

This height field is one of the Franke's data sets. We sampled the above data set on a grid size of 38x38.

We explored the effect of choosing different weights by experimenting with different values of the weights (w_1, w_2, w_3) . Figure 3 presents errors associated at different levels of compression for some of these weights. Figure 4 displays the error graphs associated with six of these weights as the compression is gradually increased.

Weights	EMD Error	Wgt. Error	Mag. Error	Ang. Error	Compression
(1,0,0)	5.84	0.12	0.12	1.34	63.3
	27.13	0.75	0.75	3.17	90.0
(0,1,0)	18.06	0.70	2.84	0.70	90.0
(1,1,0)	22.04	0.74	1.10	1.10	90.0
(1,0,10k)	0.00	1.19	1.19	2.84	90.0
(0,1,10k)	0.00	1.00	1.48	1.21	90.0
(1,1,10k)	0.00	0.78	2.89	0.78	90.0

Figure 3: Compression statistics for the Franke data set. All values are in percentages.

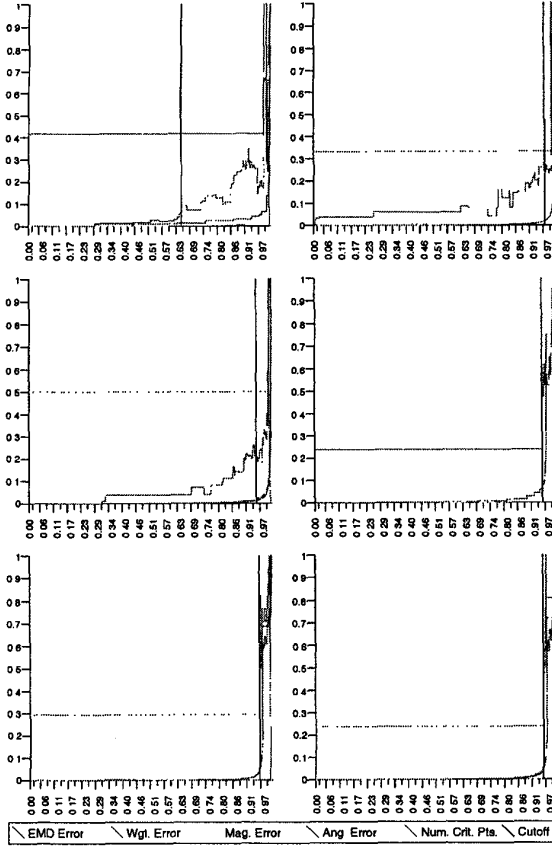


Figure 4: Error vs. compression graphs for the Franke data set with following weights: (i) upper left: (1,0,0), (ii) upper right: (0,1,0), (iii) middle left: (1,1,0), (iv) middle right: (1,0,10k), (v) lower left: (0,1,10k), (vi) lower right: (1,1,10k).

The first five images of Figure 5 show the critical points and the integral curves computed from the original and the compressed data sets.

We say a few words about the topology display in the color plate. Interior critical points are shown as red disks (spirals) if they are attracting nodes (foci respectively) and shown as green disks (spirals) if they are repelling nodes (foci respectively). Saddle points are shown as blue disks. A set of four integral curves shown in blue and red lines are computed originating from each saddle point in the directions of the eigenvectors. In addition, positive boundary switch points are shown as blue disks along the boundary. A positive boundary switch point is a point where the flow switches from outward to inward; a negative boundary switch point is a point where the flow switches from inward to outward. For further details, we refer the reader to [4]. Integral curves are also computed from the positive boundary switch points. In addition, the boundary is red where the flow is outbound and the boundary is green when the flow is inbound. For the Franke's data set, there are five critical points (2 saddles, 2 attracting nodes and 1 repelling node) in the interior and two positive boundary switch points.

For this data set, in all the cases except for (1,0,0) weight, irrespective of whether the topology preserving constraint of $w_3 = 10k$ was imposed or not, the local angular and magnitude errors are rather small all the way up to 90% compression, as shown in

Figure 3. We define the *cutoff point* as the level of compression where both the number of critical points and the type of critical points remain the same as the original topology. These cutoff points are shown as vertical grey lines in Figure 4. The cutoff point is reached for the weight (1,0,0) at 63.3% compression level. Topology generated from the compressed data is shown in the upper middle image next to the original topology shown in the upper left of Figure 5. For all the remaining weights, the cutoff compression was in the range of 92% to 95%. We observed that the degradation in the visual appearance of the integral curves very close to the cutoff point is significant. Figures 5 (d) and (e) show the topology generated after compressing the data by 90% using the weights (1,1,0) and (1,1,10k) respectively. As is evident, the topology remains unchanged at these levels of compression. For contrast, we also display the topology after 90% compression in Figure 5 (c) for the weight (1,0,0). In this case, although the number of critical points remain the same, two critical points have changed from being attracting nodes to attracting foci.

It is interesting to observe (see Figures 3 and 4 that for weights $w_3 = 10k$, EMD errors are zero all the way upto 90% compression. For $w_3 = 0$, EMD errors increase 18% to 27%. However, except in the case of the weight (1,0,0), EMD errors are attributed to the change in the α and β values associated with the critical point, while the number of critical points remain the same, as is shown by the horizontal thin red lines in Figure 4. We also know from Figure 5 that the type of critical points remain the same as well in these cases. We also explored many other values in the weight space such as (1,2,10k), (0,4,10k), and found consistent 90% or higher cutoff points, although the behavior was non-linear.

In summary, for this data set, we obtained topology preserving 90% compression with very low angular and magnitude errors and zero EMD error using the constrained ($w_3 = 10k$) cluster algorithm for most weights. We were also able to obtain topology preserving 90% compression with very low angular and magnitude errors, although with high EMD errors, even with the unconstrained ($w_3 = 0$) cluster algorithm for many weights.

4 Implementation

In this section, we present some implementation details – namely the interactive capability to investigate the effects of different levels of compression and weights on topology and error using visual display and graphical user interface.

We implemented the compression algorithm in C and use the OpenGL and XForms libraries for the SGI platform. Our graphical user interface is highly interactive and user friendly. The interface contains buttons, sliders, input fields, and counters. These interfaces facilitate user construction of various weighted error metrics, resulting in different types of compression of the data set. A visual graph provides a global view of the compression errors (EMD, weighted error, magnitude, and angular) and the number of critical points associated with each compression level. A user can step through the graph by compression level, percentage compression, or error threshold. As a user traverses the graph the errors and topology at the indicated compression level can be viewed simultaneously. Errors are also shown numerically as a user traverses the error graph.

In addition, vector glyphs are used to display the vector field. A user can better control the visual display of a dense vector field by adjusting the vector glyph length (unit, unit scale or default), adjusting the size of the critical point glyphs and modifying the width of the integral lines. Scaling, translations and rotation can also be applied to the vector field to zoom-in and isolate any particular area of interest. These features provide the user with several investigational methods to draw conclusions about compression bounds for a data set.

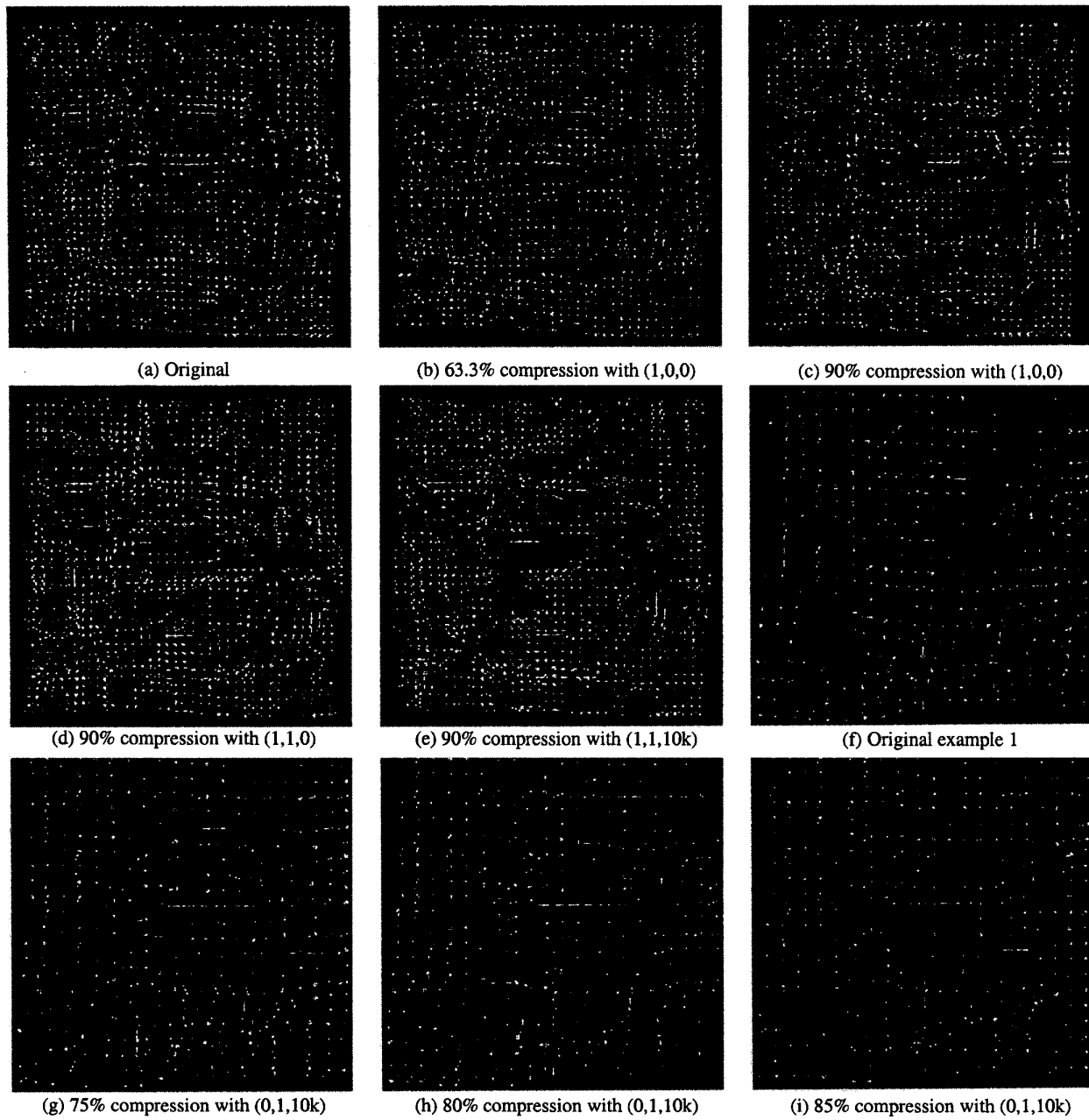


Figure 5: Topology visualization superimposed with vector glyphs at varying levels of compression. The first five images belong to the Franke data set; the last four images belong to example 1. (a) original topology has 5 interior critical points and 2 positive boundary switch points, (b) topology is preserved, but EMD error is not zero, (c) two critical points change from attracting nodes to attracting foci, (d) topology is preserved, but EMD error is not zero, (e) topology is preserved and EMD error is zero, (f) original topology with 7 interior critical points and 2 positive boundary switch points, (g) topology is preserved and EMD error is zero, (h) topology is preserved and EMD error is still zero, and (i) topology is destroyed by addition of a critical point.

5 Results

We now discuss the application of our compression algorithm to two data sets.

Example 1: Let v_x and v_y denotes the x -component and the y -component of the 2D vector field respectively. The first data set is then defined by the following equations:

$$\begin{aligned} v_x = & x^7 - x^6 + .52x^5 + .34x^4 - .6x^3 + .63x^2 - .43x \\ & + 3x^5y^2 + 1.92x^5y - .38x^4y^2 - .68x^4y^2 + 3x^3y^4 \\ & + 3.84x^3y^3 - .48x^3y^2 - 1.47x^3y + 2.24x^2y^4 \\ & + 3.43x^2y^3 + 2.25x^2y^2 + 1.85x^2y + xy^6 + 1.92xy^5 \\ & - xy^4 - 5.39xy^3 - 5.62xy^2 - 2.51xy + 1.62y^6 \\ & + 4.1y^5 + 3.68y^4 + 1.82y^3 + 1.22y^2 + .78y + .19; \\ v_y = & -1.09x^6 + 1.86x^5 - 1.56x^4 + 1.01x^3 - .67x^2 + .11x \\ & - x^6y + 2.62x^5y - 3x^4y^3 - 5.19x^4y^2 - 4.12x^4y \\ & + 5.24x^3y^3 + 8.51x^3y^2 + 5.1x^3y - 3x^2y^5 - 7.11x^2y^4 \\ & - 6.72x^2y^3 - 5.86x^2y^2 - 3.62x^2y^1 + 2.62xy^5 \\ & + 6.64xy^4 + 6.86xy^3 + 3.45xy^2 + .82xy - y^7 - 3.01y^6 \\ & - 2.6y^5 - .38y^4 - .07y^3 - .47y^2 - .17y + .02. \end{aligned}$$

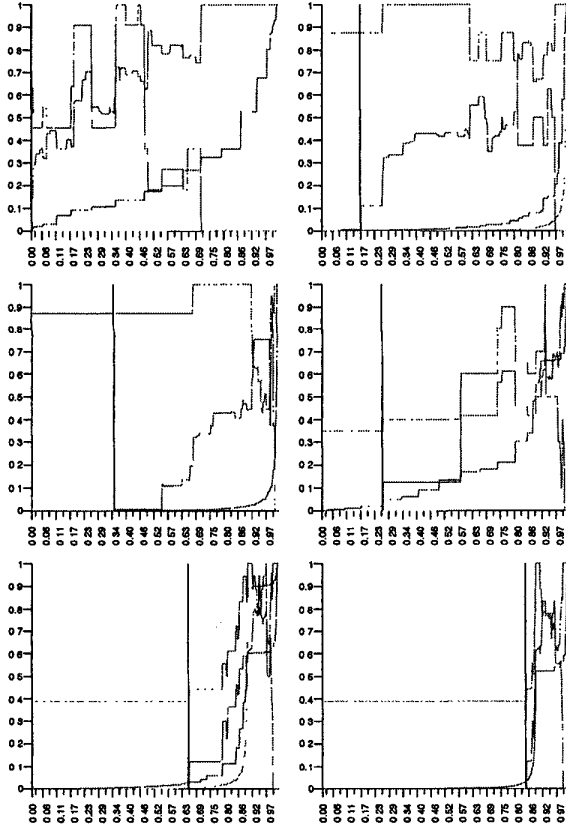


Figure 6: Error vs. compression graphs for data set in example 1 with following weights: (i) upper left: (1,0,0), (ii) upper right: (1,1,0), (iii) middle left: (0,1,0), (iv) middle right: (1,0,10k), (v) lower left: (1,1,10k), (vi) lower right: (0,1,10k).

Weights	EMD Error	Wgt. Error	Mag. Error	Ang. Error	Cutoff Point
(1,0,0)	0.00	0.00	0.00	0.00	0.00
(1,1,0)	0.04	0.03	0.02	0.21	15.9
(0,1,0)	0.73	0.25	10.55	0.25	53
(1,0,10k)	0.00	0.01	0.01	1.92	24.8
(1,1,10k)	0.00	0.48	0.55	2.68	64.1
(0,1,10k)	0.00	3.43	42.46	3.43	83.2

Figure 7: Compression statistics for the data set in example 1. All values are percentages.

This seventh degree polynomial vector field is obtained by unwinding the Clifford algebra expression for the polynomial vector field used in [12]. Figure 7 presents errors associated at different levels of compression for six different weights. Figure 6 displays the error graphs associated with these weights as the compression is gradually increased. The last four images of Figure 5 show the critical points and the integral curves computed from the original and the compressed data sets using the weight (0, 1, 10k) at different levels of compression.

The topology program captures 7 critical points in the interior and 2 positive boundary switch points for this data set as shown in the middle right image of Figure 5. Furthermore, in this case, three of the critical points are very close to each other. For this data set, the cutoff points using unconstrained cluster algorithm ($w_3 = 0$) allow very little compression. In fact, for the weight (1, 0, 0), the type of a critical point changes in the very first step as shown in the upper left graph of Figure 6. Even for the other two weights (1, 1, 0) and (0, 1, 0), the cutoff points allow 15.9% and 53% compression only, as shown in Figure 7. Figure 6 shows that at these levels of compression, although the number of critical points remain the same, the type of critical point changes. Even for the unconstrained weights, with weight (1, 0, 10k), the cutoff compression is only 24.8%. The cutoff compression of 64.1% using the weight (1, 1, 10k) is good, although a much better 83.2% compression is achieved using the weight (0, 1, 10k) while keeping the EMD error at zero and local magnitude and angular errors still rather small. For this weight, we have displayed the topology obtained by compressing the data set by 75%, 80% and 85% respectively in the lower left, lower middle and the lower right images of Figure 5. The first two of these are topology preserving compression, while the last one (past the cutoff point) introduces an additional saddle point destroying the topology as well as making the integral curves appear quite choppy.

In summary, this data set is sensitive to the choice of weights. Significant compression preserving topology can still be achieved with low errors by using a constrained cluster algorithm ($w_3 = 10k$) and giving high weight to the local angular error.

Example 2: For the final example, we use the skin friction data set obtained by direct numerical simulation of a turbulent flow around a square cylinder. The data was generated by R.W.C.P. Verstappen and A.E.P. Veldman of the University of Groningen (the Netherlands) and is the same data set used in [4]. The technique used to generate the data is described in [14]. The skin friction is computed on one of the four sides of the cylinder resulting in a data set with a resolution of 104×64 .

For the skin friction data set, we present the results by picking a promising representative from each of the unconstrained and constrained cluster algorithm from the weight space that we explored. These representative weights are (1, 1, 0) and (1, 1, 10k). Figure 11 presents errors associated with and the number of critical points at different levels of compression for these two weights. Figure 10 displays the error graphs associated with these two weights as the compression is gradually increased. Images of Figures 8 and 9

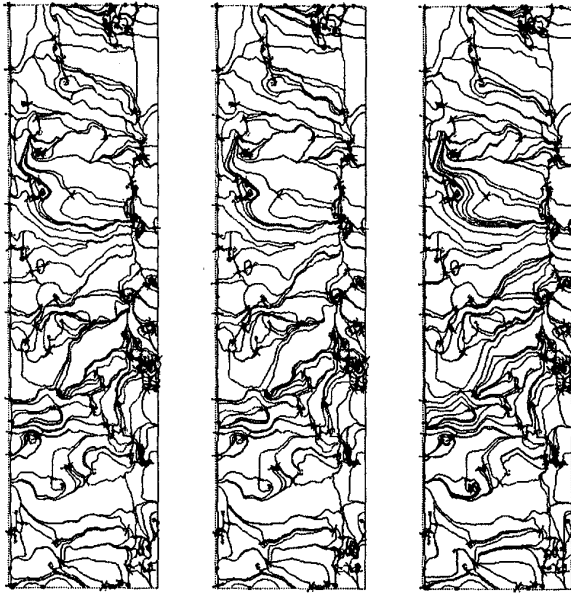


Figure 8: Skin friction dataset: (a) Original (b) 55% compression with weights (1,1,10k) (c) 65% compression with weights (1,1,10k)

show the critical points and the integral curves computed from the original and the compressed data sets.

The skin friction data set has 338 critical points in the interior. The original topology is shown in the left image of Figure 8. This data set has a large number of critical points many of them very close together. For this data set, the cutoff points using the weights (1,1,0) and (1,1,10k) produce 4.7% and 37.6% compression only. As can be seen in the left graph of Figure 9, critical points (downward sloping thin red line) start getting eliminated at the very beginning using the unconstrained algorithm. The pace of the elimination of the critical points is slow almost all the way up to 65% compression. Even though the original topology is destroyed, the EMD error is not too large (8% approximately). Surprisingly, the local magnitude and angular error remains small all the way up to 95% compression. Topology obtained at 55%, 65% and 95% compression levels using the unconstrained cluster algorithm with the weight (1,1,0) is shown in Figure 9. Even at high levels of compression, it seems that some of the major features such as attachment and separation lines seem to remain intact. This example seems to suggest that the unconstrained cluster algorithm can perhaps be used for simplifying (non-preserving) topology.

For the constrained algorithm with the weights (1,1,10k), the cutoff compression is 37.6% as mentioned before, as is indicated by a hardly noticeable jump in the thin horizontal line in the right graph of Figure 10. However, the compression can be increased all the way up to 51.3% with an addition of 2 more critical points (while keeping all the original critical points). Topology obtained by compressing the data set by 55% and 65% using the weights (1,1,10k) are shown in the middle and the right images of Figure 8. For this example, topology preserving compression ratio (37.6%) is rather small; however, significant compression (60%) can be achieved with low angular, magnitude and topology error as measured by the EMD.

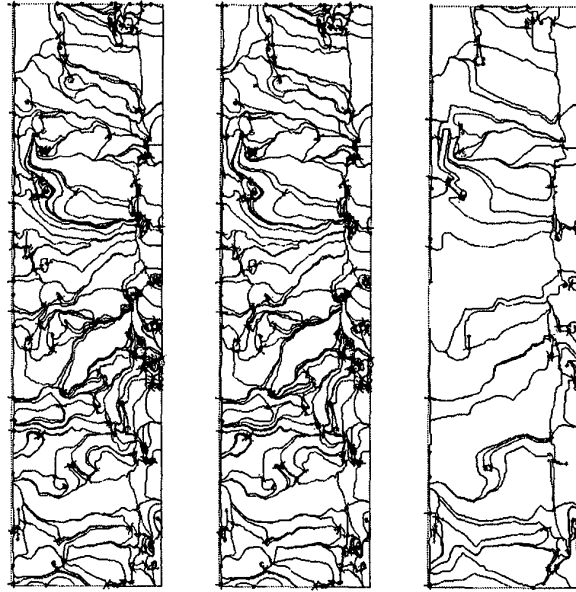


Figure 9: Skin friction dataset: (a) 55% compression with weights (1,1,0) (b) 65% compression with weights (1,1,0) (c) 95% compression with weights (1,1,0)

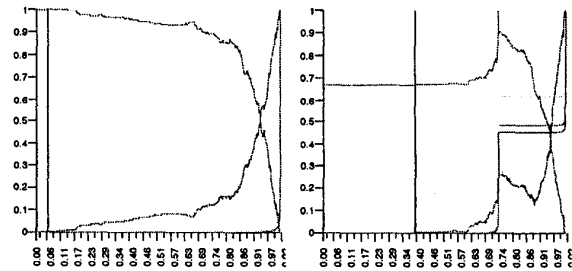


Figure 10: Error vs. compression graphs for the skin friction data set with following weights: (i) left: (1,1,0), (ii) right: (1,1,10k).

Weights	EMD Error	Wgt. Error	Mag. Error	Ang. Error	Crit. Pts.	Compression
(1,1,0)	.26	0.00	0.00	0.00	338	4.7
	1.21	0.01	0.01	0.00	339	12.8
	8.30	0.02	0.08	0.03	316	55.0
	7.94	0.03	0.09	0.03	317	60.0
	8.69	0.03	0.13	0.04	315	65.0
	67.44	4.90	0.91	0.52	113	95.0
(1,1,10k)	0.00	0.01	0.04	0.01	338	37.6
	0.58	0.02	0.11	0.03	340	51.3
	1.16	0.03	0.14	0.04	342	55.0
	2.87	0.06	0.22	0.07	348	60.0
	4.78	0.13	0.34	0.14	355	65.0

Figure 11: Compression statistics for the skin friction data set. All values (except the number of critical points) are percentages.

6 Discussion and Conclusions

We have presented a compression algorithm for 2D vector fields that can preserve topology while maintaining low magnitude and angular errors. We have been able to achieve 90% topology preserving compression in a data set where there are very few critical points. In another data set, where critical points are close together but not too many, we have been able to achieve 80% topology preserving compression. In a third data set, characterized by a large number of critical points, we have been able to achieve approximately 37% compression while preserving topology or 50% compression with a very small EMD error.

We have also provided a visual display and interactive environment for users to experiment with new error metrics and view compression statistics, vector clusters and topology. Our compression algorithm can easily be enhanced to deal with higher order critical points as well.

Although topology is an important feature of fluid flow, presence of a large number of critical points can diminish the utility of a topology visualization by creating a cluttered display [4]. In these cases, the compression algorithm presented in this paper can also be used to simplify the topology by reducing the number of critical points. To this purpose, it may be judicious to assign different weights w_3 (varying levels of importance) to critical points – lower weights to the critical points with lesser influence and higher weights to critical points with higher influence. The influence of a critical point can be measured in terms of flow area or the distance to the nearest critical point or by a new metric. We plan to investigate this approach in a future work. Alternatively, one can use our topology preserving compression algorithm as the first pass of a two-pass compression algorithm. The second pass algorithm can then focus on simplifying the topology using a critical point influence criteria.

There are many other interesting directions of investigations that this research has opened up. Here we mention two such possibilities – a new global error metric and a top-down compression algorithm. EMD measures the distance between two vector fields simply based on the number, type and phase plane coordinates of the critical points. What about a new metric that takes into account the locations of critical points, properties of integral curves or additional features of the flow? Another possibility is to explore a top-down compression algorithm. Our bottom-up compression algorithm precomputes the topology of the original data set. What if this luxury is not available? A top-down compression algorithm will start with meagre data and add details to the data set successively. Can such an algorithm determine that the original topology (or close to the original topology) is captured so that there is no need to add additional details to the data set? We hope to pursue some of these research directions as well.

7 Acknowledgments

This work was partially supported by LLNL Agreement No. B347879 under DOE Contract No. W-7405-ENG-48 and the Multi-disciplinary Research Initiative (MURI) grant by DOD. For the use of the topology generation code and the skin friction data set, we would like to thank Wim de Leeuw. For the use of the EMD code, available freely from their web site, we would like to thank Rubner and Tomasi. We would also like to thank Jonathan Cheatham for helping us with parts of this project. Special thanks go to Wim de Leeuw for many helpful discussions.

References

- [1] C. L. Bajaj, V. Pascucci, and D. R. Schikore. Visualization of scalar topology for structural enhancement. In *Proceedings: Visualization '98*, pages 51–58. IEEE Computer Society, October 1999.
- [2] C. L. Bajaj and D. R. Schikore. Topology preserving data simplification with error bounds. *Journal on Computers and Graphics*, 22(1):3–12, 1998.
- [3] R. K. Batra and L. Hesselink. Feature comparisons of 3d vector fields. In *Proceedings: Visualization '99*, pages 105–114. IEEE Computer Society, October 1999.
- [4] Wim C. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In *Proceedings: Visualization '99*, pages 349–354. IEEE Computer Society, October 1999.
- [5] Al Globus, Creon Levit, and Tom Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *Proceedings: Visualization '91*, pages 33 – 40. IEEE Computer Society, 1991.
- [6] B. Heckel, G. H. Weber, B. Hamann, and K. I. Joy. Construction of vector field heirarchies. In *Proceedings: Visualization '99*, pages 19–26. IEEE Computer Society, October 1999.
- [7] J. L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, May 1991.
- [8] D. Kenwright. Automatic detection of open and closed separation and attachment lines. In *Proceedings: Visualization '98*, pages 151–158. IEEE Computer Society, 1998.
- [9] Y. Lavin, R. K. Batra, and L. Hesselink. Feature comparisons of vector fields using earth mover's distance. In *Proceedings: Visualization '98*, pages 103–109. IEEE Computer Society, October 1998.
- [10] Y. Rubner and C. Tomasi. The earth mover's distance as a metric for image retrieval. Tech Report STAN-CS-TN-98-86, Department of Computer Science, Stanford University, September 1998.
- [11] G. Scheuermann, H. Hagen, H. Kruger, M. Menzel, and A. Rockwood. Visualization of higher-order singularities in vector field. In *Proceedings: Visualization '97*, pages 67–74. IEEE Computer Society, October 1997.
- [12] G. Scheuermann, X. Tricoche, and H. Hagen. C1-interpolation for vector field topology visalization. In *Proceedings: Visualization '99*, pages 271–278. IEEE Computer Society, October 1999.
- [13] A. Telea and J. J. van Wijk. Simplified representation of vector fields. In *Proceedings: Visualization '99*, pages 35–42. IEEE Computer Society, October 1999.
- [14] R.W.C.P. Verstappen and A.E.P. Veldman. Spectro-consistent discretization of navier-stokes: a challenge to rans and les. *Journal of Engineering Mathematics*, 34:163–179, 1998.
- [15] J. Wilhelms and A. Van Gelder. Multi-dimensional trees for controlled volumetric rendering and compression. In *ACM Symposium on Volume Visualization '94*, pages 27–34, 1994.